IEEE Test and Diagnostics Standards

John Sheppard
ARINC
2551 Riva Road
Annapolis, MD 21401
410-266-2099
jsheppar@arinc.com

Mark Kaufman
NWAS
PO Box 5000
Corona, CA 91718
909-273-5725
kaufman.mark@corona.navy.mil

Test And Diagnosis Standards

- **IEEE Std 1232-1995.** *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Overview and Architecture*
- **IEEE Std 1232.1-1997.** *IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Data and Knowledge Specification*
- **IEEE Std 1232.2-1998.** *IEEE Trial-Use Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE): Service Specification*
- **IEEE Std P1522.** *Draft Standard for Standard Testability and Diagnosability Characteristics and Metrics*

NDIA NATIONAL DEFENSE INDUSTRIAL ASSOCIATION

Diagnosis

- Derived from two Greek words:
  - διά: about/through
  - γιγνοσκην: discernment/knowledge
- Any conclusion that can be drawn about the health state of a system under test.
- Includes “no fault.”

Information Model

- An information model is a formal description of types (classes) of ideas, facts, and processes that together form a model of a portion of interest of the real world.
- Information models provide a formal specification of the *semantics* of information in an “Information System”

Information Model

- **Purpose:** To identify clearly the objects in a “domain of interest” to enable precise communication about that domain.
- **Comprises**:
  - objects or entities
  - relationships
  - constraints
- When taken together, these provide a complete, unambiguous, formal representation of the information in the domain of interest.

Information Exchange Files

- Information can be stored in files by one application and read from the files by another.
- The file format provides a common syntax.
- The legal content of the file is defined by the semantics of the model.

Information Exchange Services

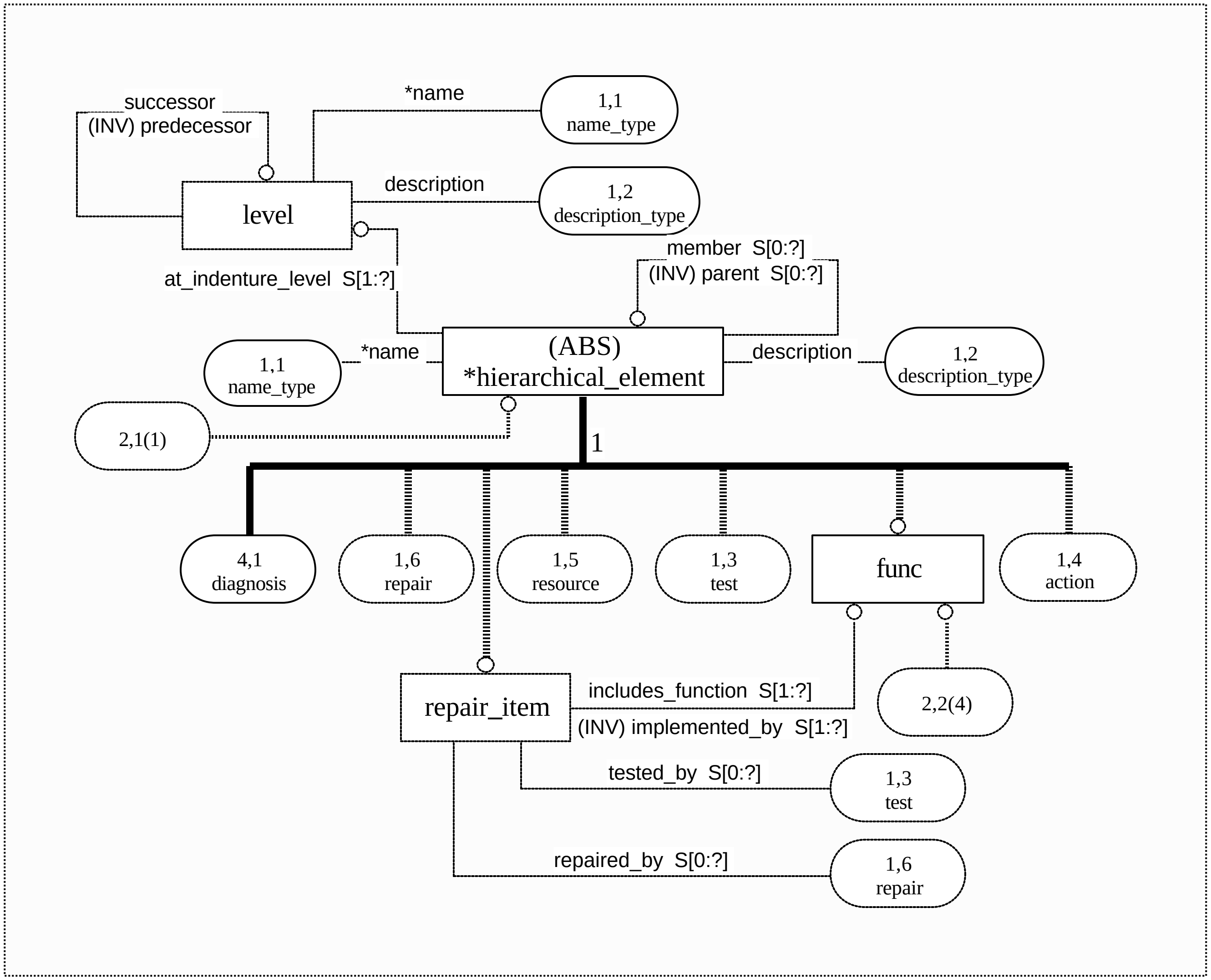
- Information can be shared between applications by way of software or hardware services over a communications backbone.
- The interface definition provides a common syntax.
- The legal content of the message is defined by the semantics of the model.

NDIA NATIONAL DEFENSE INDUSTRIAL ASSOCIATION

Common Element Model

- Contains model elements in common to diagnosis independent of diagnostic approach.
- Provides for hierarchical relationships among model elements.
- Includes model for cost attributes.
- Captures information about required context.
- Developed in ISO 10303 Part 11 (EXPRESS).

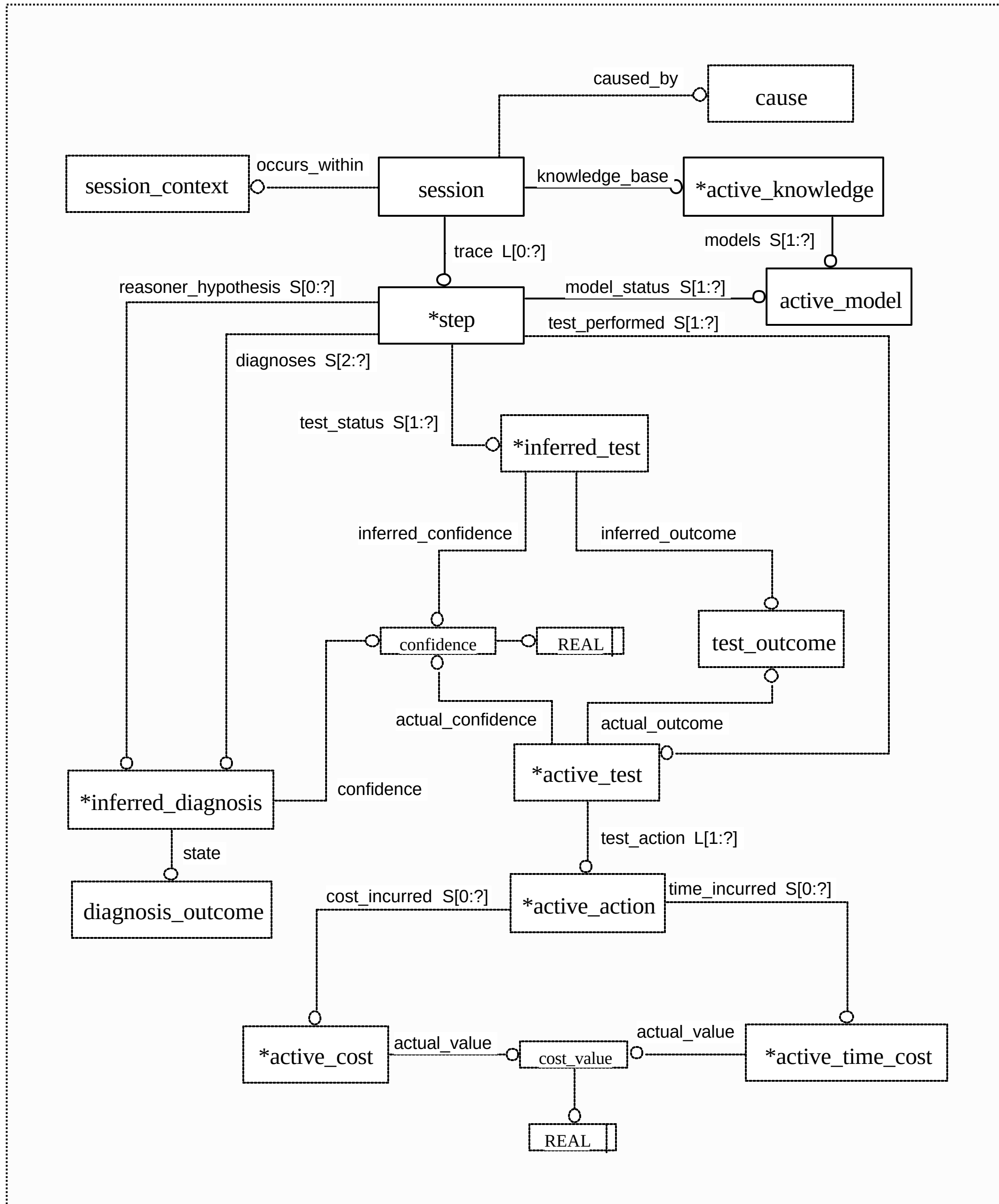
Simplified CEM



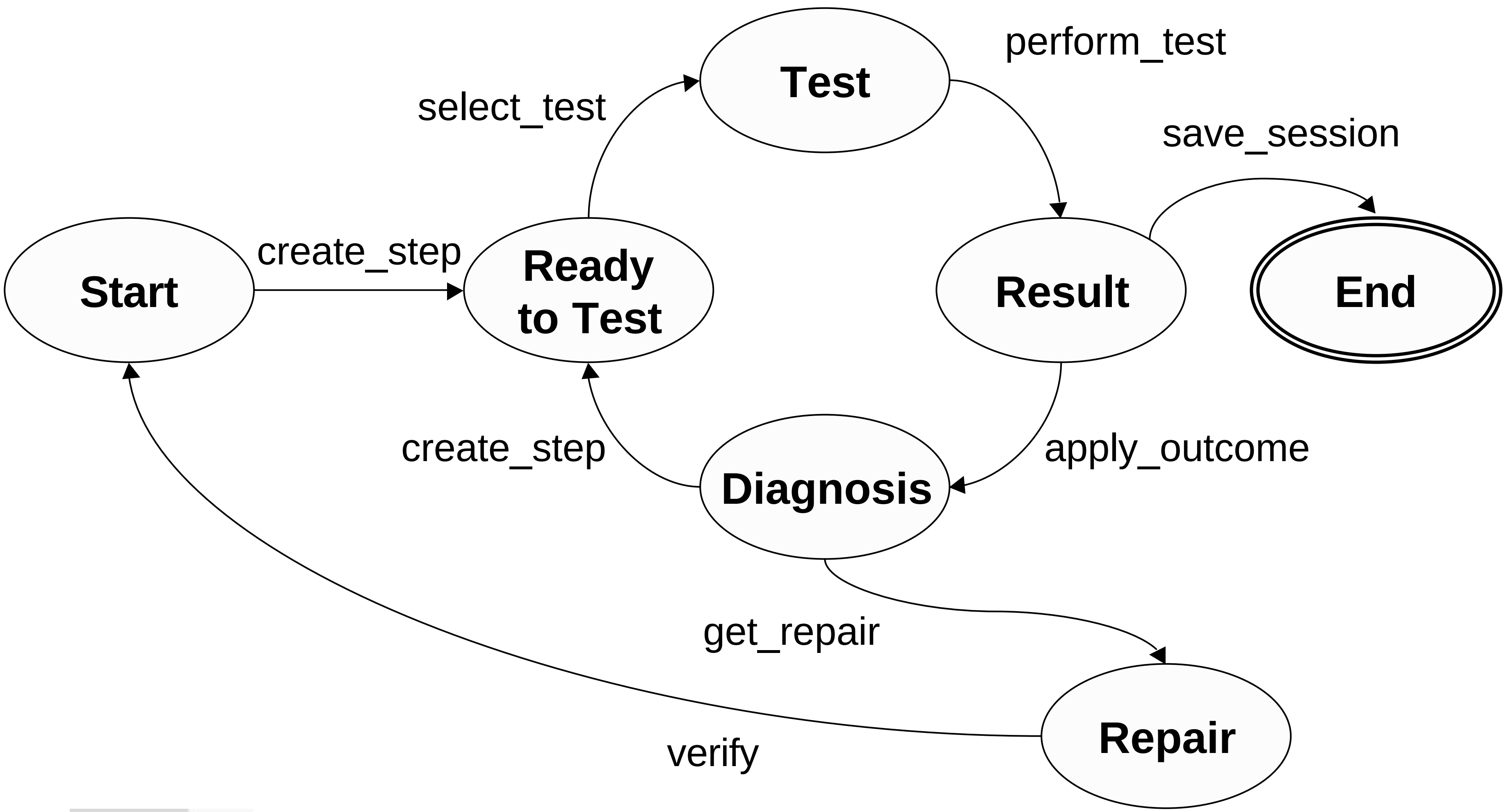
Dynamic Context Model

- Captures state of diagnostic reasoning.
- Compatible with all of the inference models defined within AI-ESTATE.
- Instantiated during a diagnostic session.
- Can be saved to provide historical trace of diagnostic process.
- Includes current context for comparison with required.
- Developed in ISO 10303 Part 11 (EXPRESS).

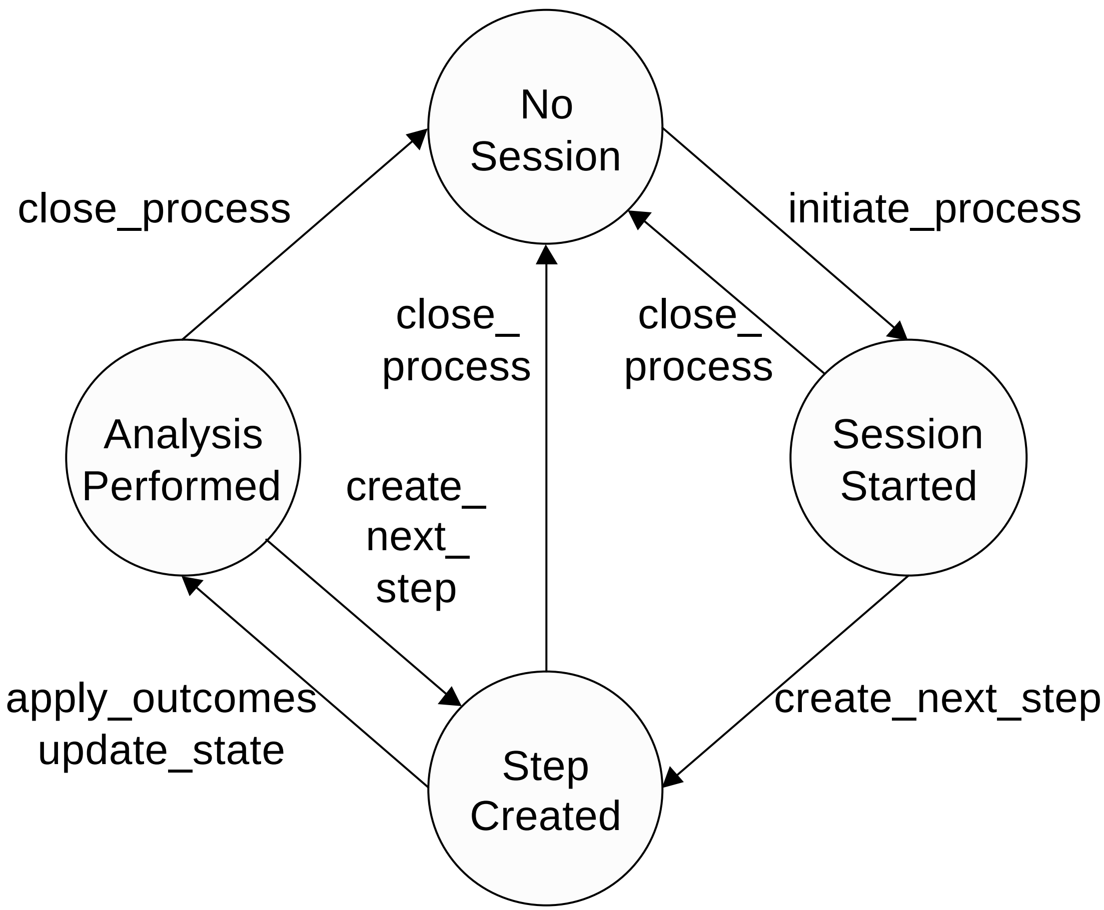
Simplified DCM



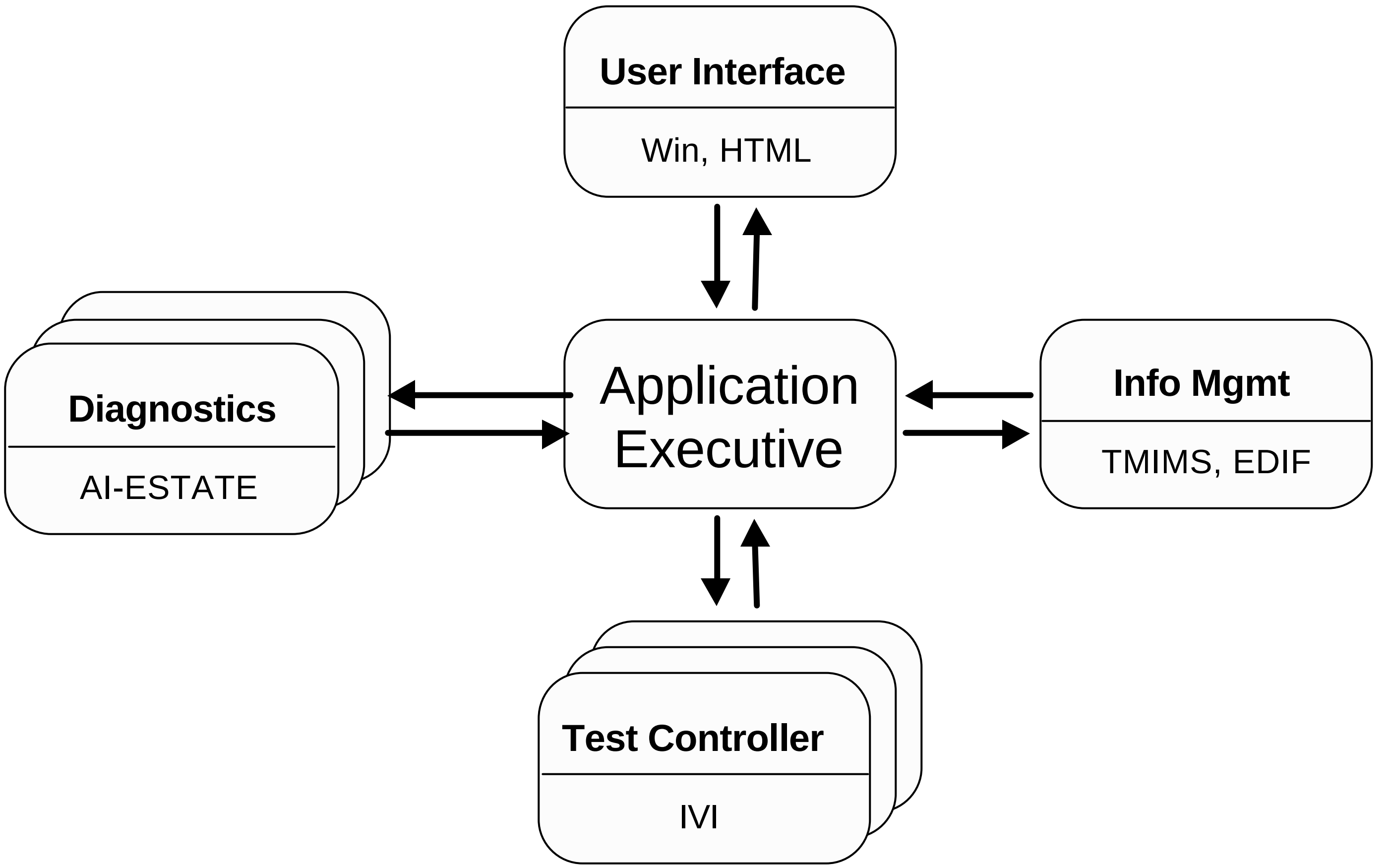
General Diagnostic Process



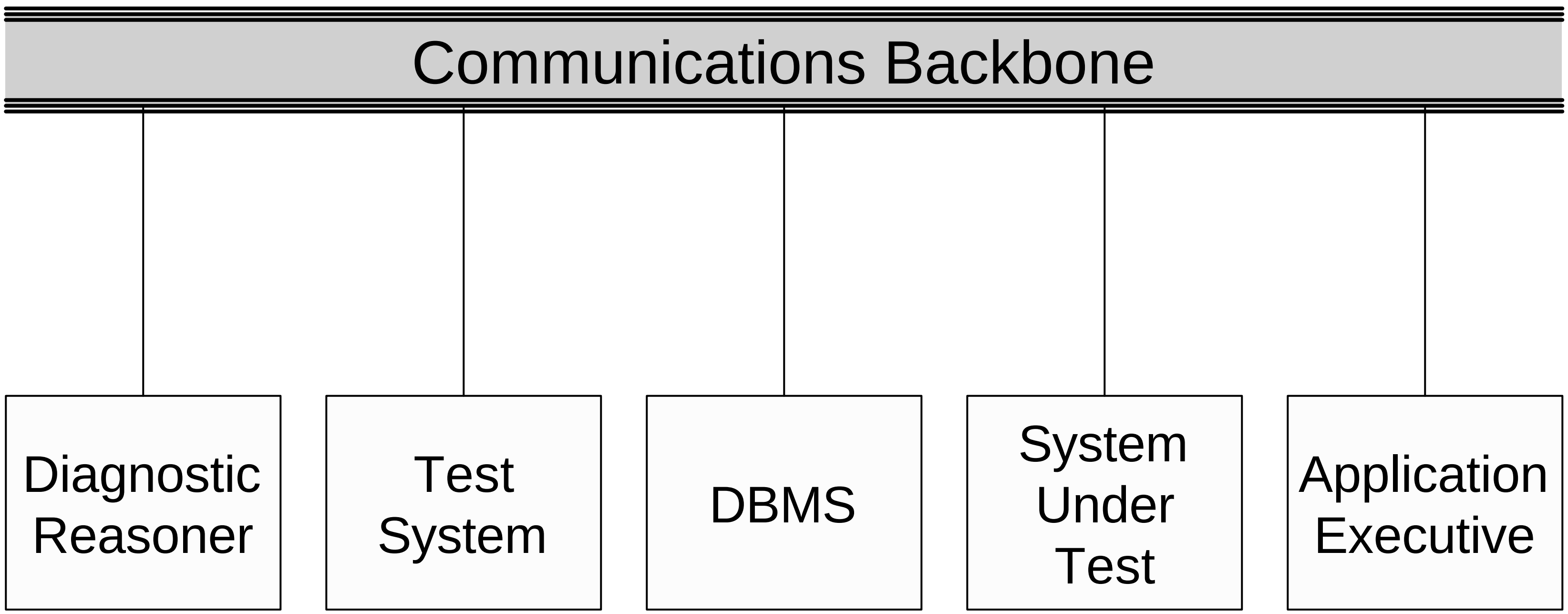
AI-ESTATE Execution Model



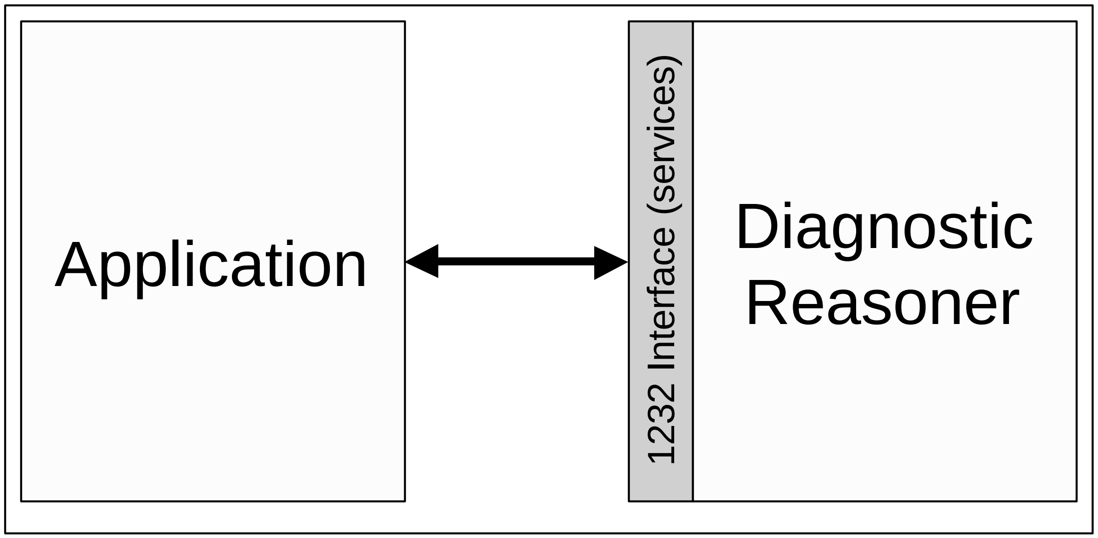
AI-ESTATE Client-Server View



AI-ESTATE Component View



AI-ESTATE Application Model



Service Definitions

- Use EXPRESS to define function and procedure prototypes.
- All services one of `create`, `put`, `get`, `delete`.
- Service categories:
  - Model traversal services
  - Reasoner control services
  - Utility and counting services
- Set of higher-order services defined.

Testability

- A design characteristic which allows the status (operable, inoperable, or degraded) of an item to be determined and the isolation of faults within the item to be performed in a timely manner.

MIL STD 2165

Testability

- An Equipment Has Good Testability If Faults Can Be *Confidently* and *Efficiently* Identified.
  - Confidently Means Frequently and Unambiguously Identifying Only Failed Elements With No Removal of Fault-tree Elements.
  - Efficiently Means Optimizing the Resources Required.

Diagnosability

- The ability to discern the health state of the system.
- Testability and Diagnosability are intrinsically related.
  - Cannot diagnose without tests.
  - Testing without diagnosis is a vacuous endeavor.

Metrics, Why Worry?

- Terms Not Precisely Defined or Have Multiple Definitions
- Different Diagnostic Tools Calculate Terms Differently
- Establishing Requirements, and Predicting and Evaluating Testability Are Difficult

NDIA NATIONAL DEFENSE INDUSTRIAL ASSOCIATION

Metrics Goals

- Precise and Unambiguous Definitions
- Precise and Unambiguous Calculations
- Derived from Test Model
  - Not an Isolated Definition
- Repeatable Metrics

Two 2165 Definitions of FFD

- The fraction of all faults detected by BIT and external test equipment
- The fraction of all detectable faults detected by BIT and external test equipment

NDIA
NATIONAL DEFENSE INDUSTRIAL ASSOCIATION

More FFD Definitions

- Fraction of all faults detected through the use of defined means. Defined means implies all means of detection that have been identified.
- Percentage of all faults automatically detected by BIT/ETE
- Percentage of all faults detectable by BIT/ETE

More FFD Definitions

- Percentage of all faults detectable on-line by BIT/ETE
- Percentage of all faults and out-of-tolerance conditions detectable by BIT/ETE
- Percentage of all faults detectable by any means

Precise FFD

$$FFD_{level} = {num\_detectable_{level}} \big/ {num\_diagnosis_{level}}$$

Example (num_diagosis)

```
FUNCTION num_diagnosis(model:EDIM.edim; lvl:CEM.level) : NUMBER;
        LOCAL
                diag_count : NUMBER;
        END_LOCAL;
        diag_count := SIZEOF(QUERY(tmp <* model.model_diagnosis |
                                tmp.level_of_diagnosis = lvl);
        RETURN(diag_count);
END_FUNCTION;
```

Example (num_detectable)

```
FUNCTION num_detectable(model:EDIM.edim; lvl:CEM.level) : NUMBER;
        LOCAL
                diags : SET [0:?] OF EDIM.inference
                detect_set : SET [0:?] OF CEM.diagnosis := NULL;
        END_LOCAL;
        REPEAT I := LOINDEX(model.inference) TO HIINDEX(model.inference);
                diags := QUERY(tmp <* model.inference[I].conjuncts |
                        (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
                diags := diags + QUERY(tmp <* model.inference[i].disjuncts |
                        (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
                diags := QUERY(tmp <* diags |
                        tmp.pos_neg = negative OR
                        NOT(tmp.diagnostic_assertion = 'Good'));
                detect_set := detect_set +
                        QUERY(tmp <* diags.for_diagnosis |
                                tmp.level_of_diagnosis = lvl);
        END_REPEAT;
        RETURN(SIZEOF(detect_set));
END_FUNCTION;
```

Fraction of Faults Detected

```
FUNCTION ffd(model:EDIM.edim; lvl:CEM.level) : REAL;
        LOCAL
                diag_count : INTEGER;
                diags : SET [0:?] OF EDIM.inference
                detect_set : SET [0:?] OF CEM.diagnosis := NULL;
        END_LOCAL;
        diag_count := SIZEOF(QUERY(tmp <* model.model_diagnosis |
                        tmp.level_of_diagnosis = lvl);
        REPEAT I := LOINDEX(model.inference) TO HIINDEX(model.inference);
                diags := QUERY(tmp <* model.inference[I].conjuncts |
                        (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
                diags := diags + QUERY(tmp <* model.inference[i].disjuncts |
                        (TYPEOF(tmp) = 'EDIM.diagnostic_inference'));
                diags := QUERY(tmp <* diags |
                        tmp.pos_neg = negative OR
                        NOT(tmp.diagnostic_assertion = 'Good'));
                detect_set := detect_set +
                        QUERY(tmp <* diags.for_diagnosis |
                                tmp.level_of_diagnosis = lvl);
        END_REPEAT;
        RETURN(SIZEOF(detect_set)/diag_count);
END_FUNCTION;
```

Primitives

- Full EXPRESS specification, though formal and unambiguous, is difficult to read.
- Most metrics can be specified in terms of formally defined primitives.
- Primitives are defined in EXPRESS.
- Metrics are defined using standard mathematical notation with primitives as constituent terms.

Candidate Set of Primitives

- Number of functions
- Number of faults
- Number of detectable faults
- Number of non-detectable faults
- Test Cost
- Test Confidence
- Number of tests
- Number of Units (LRU, SRU, etc)
- Number of Isolatable Units
- Repair Cost
- Replacement Cost
- Failure Rate

Summary

- Sharing information is key to *any* process.
- Formal models are required to ensure information communicated is unambiguous and understood.
- Standard interfaces and models provide basis for establishing agreement on information meaning.
- P1232 provides standard information interfaces for diagnostic applications.
- P1522 provides standard, formal definitions for metrics assessing system testability/diagnosability

Contact Information

- Dr. John Sheppard, ARINC
  - 410-266-2099
  - jsheppar@arinc.com
- Mark Kaufman, NWAS
  - 909-273-5725
  - kaufman.mark@corona.navy.mil

- http://grouper.ieee.org/groups/1232